Modular Dreams Incorporated

MD+F Web Animation Kit

S/N B045387005296722-7851

SX2 REG.. 824120315736

2204

319973545333373-108

Modular Dreams Incorporated

MD+F Web Animation Kit

User Guide Version 1.0

A MD+F SX SOFTWARE

MODULAR DREAMS INC

©1997 Modular Dreams Incorporated. All Rights Reserved.

MD+F Web Animation Kit (WebAK) version 1.0 manual.

This manual, as well as the software described in it, is furnished under license and may only be used or copied in accordance with the terms of such license. The information in this manual is for informational use only, is subject to change without notice, and should not be construed as a commitment by Modular Dreams Incorporated. Modular Dreams Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in this book. Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without prior written permission of Modular Dreams Incorporated.

Unless otherwise specified, contains VisualAge libraries; Copyright © 1996 International Business Machines; used under license or sublicense. Unless otherwise specified, contains an implementation of the LZW algorithm; used under license or sublicense.

Modular Dreams Incorporated, P.O. Box 682, Vienna, VA 22183.

Printed in the USA.

Part number MDFWAKIUM10 (5/97)

Contents

Introduction

Welcome to **Modular Dreams MD+F Web Animation Kit**. This program has been designed to provide tools and features for creation of exotic elements for World Wide Web designers as well as provide means of storing the environment used to create them. You can be a graphics artist or a computer scientist; you will still find this program giving you the tools needed for the most difficult web animations.

MD+F WebAK was build to provide a easy tools ranging from an easy to use GIF Animator which you can simply drag-and-drop your image frames all the way to a fully multitasking C++ style language. And with its exclusive Collapse and Resurrect feature, you can share your creations with others like never before.

About this manual

The *MD+F WebAK User Guide* provides information on using the MD+F WebAK program as well provide simple tutorials for the WebAK scripting language. For more detailed information on the scripting features of this program, please obtain the *MD+F WebAK Programmer's Guide* which is available in the electronic format from Modular Dreams Inc's World Wide Web home page. You can also find more animation samples or collapsed environments (described elsewhere in this manual) at our Web site.

This manual assumes you have knowledge of World Wide Web page design features such as GIF Animations or Imagemaps but does provide a brief description of Word Wide Web items when they are first discussed. If you need more information on such subjects, please drop by our site's Support page. That page provides links to many sites which describe World Wide Web page design features as well as various Web Browsers for both novice and more experienced users. You will also find some interesting GIF animations there which you can grab and load to learn new tricks or to reuse on your own web pages.

Learning to use WebAK

If you are familiar with any other Modular Dreams' SX based graphics program, you should be able to become accustomed to MD+F WebAK very easily. All MD+F SX programs provide the same features such as the versatile Image View Canvases or the same powerful interface to filtering or operations windows. If not, we suggest you read the chapters in this book in

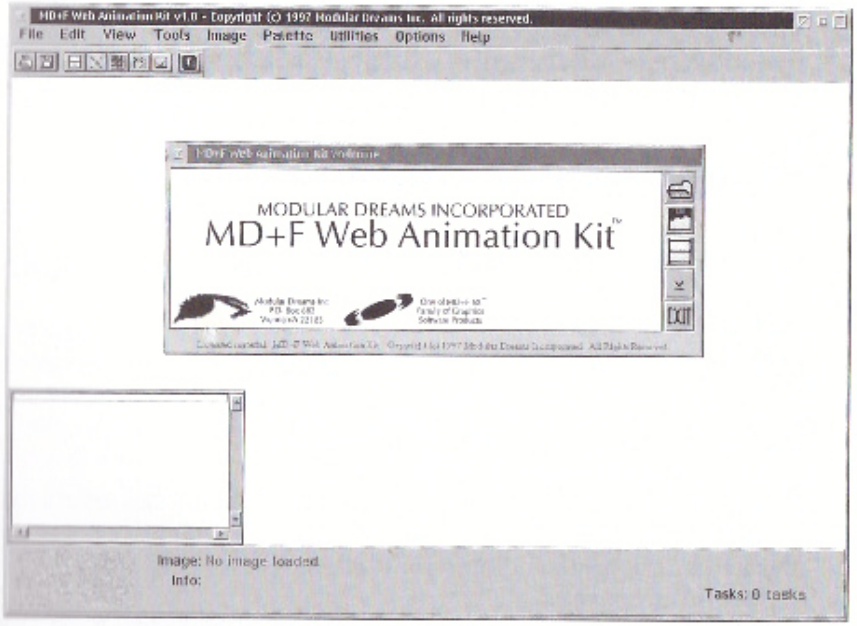
the order they are provided. And you may with to skip more specific subjects which you may not care. Since this program provides a consistent user interface on all of its tools, once you have learned one, many parts of other tools will already be familiar to you.

Using examples

Many of the examples given in this manual are accompanied by collapsed environment files. (.sxo SX object file) To load any of these files for viewing you will need to select the **File>Objects>Resurrect...** menu item and select the file or enter its path and name. Once you do this and press OK, the windows and images described in the example will be opened and displayed for you.

Chapter 1: Starting WebAK

Once you run the MD+F WebAK program, you will be presented with the WebAK main window. This large window contains menus and buttons which let you start various WebAK tools as well as an information area, at the bottom of the window, where it will display important and useful information on the actions which are taking place. The toolbar on top as well as the information area at the bottom are both optional. You may wish to disable one or both for more working area.



The controls on the main window are:

Menu bar - The application menu bar lets you start new windows, change options, etc.

Working area - This is the large white area which will contain your windows and dialogs.

Current image views - These two small images - at the bottom left hand corner - let you view the image you are working on as well as its Alpha channel. They are great for letting you know which window any operations you are performing will effect as well as give you the full picture if

you are just working on a part of it.

Image and Info status lines - These two status lines display information on the image and the operation you are performing.

Tasks status - The tasks status display shows you how many short-term tasks are in progress.

Should you wish, to disable or change the placement of the tool bar or the information bar, select the **Options>General...** menu item for the General Options window. You can also drag the toolbar to float it or to move it to one of the other borders.

Welcome Window

In addition to the main window, the Welcome window will be displayed the first time you start the application. The Welcome window gives you quick access to the more popular actions you may perform, as well as give you a way to quickly exit WebAK.

The controls on the Welcome window are:

Image Editor - This button will display a file dialog. You can then select a file and have it be displayed in an Editor Window.

Resurrect - This button will allow you to resurrect a previously collapsed WebAK environment.

GIF Animator - For easy GIF animations simply select this button. It will display the GIF Animator which you can load, view, or save GIF animations with.

Close - Select this button to close the Welcome window.

Exit - This button will close the WebAK application.

If you wish, you can configure WebAK so that the Welcome dialog is automatically removed once the application has started and loaded all its subsystems. To do this, select the **Options>General...** menu item for the General Options window.

Chapter 2: Image View Canvas

One of the visual items which you will see and interact with on many of MD+F WebAK's windows and dialogs is the Image View Canvas. They are everywhere: They can be found and are used on the simplest tools and all the way to the GIF Animator and Pattern Generator windows.

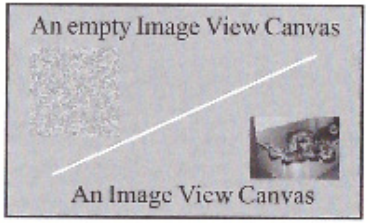


Image View Canvases are very powerful objects which you must learn to use before you can use most of WebAK's tools. Fortunately, that is an easy thing.

What is an Image View Canvas

An Image View Canvas is exactly what its name says: A canvas which lets you view an image.

Image View Canvases are:

- Usually very small.
- Shown singly or in pairs. When in pairs, the pair display both an Image's image data and its Alpha channel.
- Displaying an image. When an Image View Canvas is showing an image, it will scale it and show a 70 by 70 or a 140 by 140 representation of it. If an Image View Canvas does not contain an image, it will display static. If an Image View Canvas is showing an image, it may be showing the same image as another Image View Canvas. If that is the case, they are said to be linked.

Image View Canvases are used to:

- Represent images which are sources to or results of image processing operations.
- Hold images. This prevents cluttering of your work area.
- Show individual frames in an animation.

as well as in other places where images need to be represented.

Using an Image View Canvas

There are two ways to use an Image View Canvas:

1 - You can use the mouse to pop-up an Image View Canvas' menu and select a menu item, or
2- You can drag and drop its content.

Using an Image View Canvas' pop-up menu

An Image View Canvas' pop-up menu lets you load images into it, create other views into it, display different channels (Image or Alpha) or close the image it hold. The menu items are:

Open Image - To open a new image into the Image View Canvas, select this menu item. It will display a File dialog which you can use to select your file.

Shadow Image - When you shadow an image, you create another view into it. Select this item to create an Editor Window displaying the image.

Clone Image - When you close an image, you create another copy of it. Select this item to create an Editor Window displaying another copy of the image.

Show>Raster - Select this menu item to show the "Image" channel of your image. The image channel is what is normally shown. It is a 1 to 8 bit indexed, 24 bit, or 48 bit image.

Show>Alpha - Select this menu item to show the "Alpha" channel. The Alpha channel is a 1 or 2 byte image channel which specifies the opacity of the image. In WebAK, any Alpha channel value other than full opacity

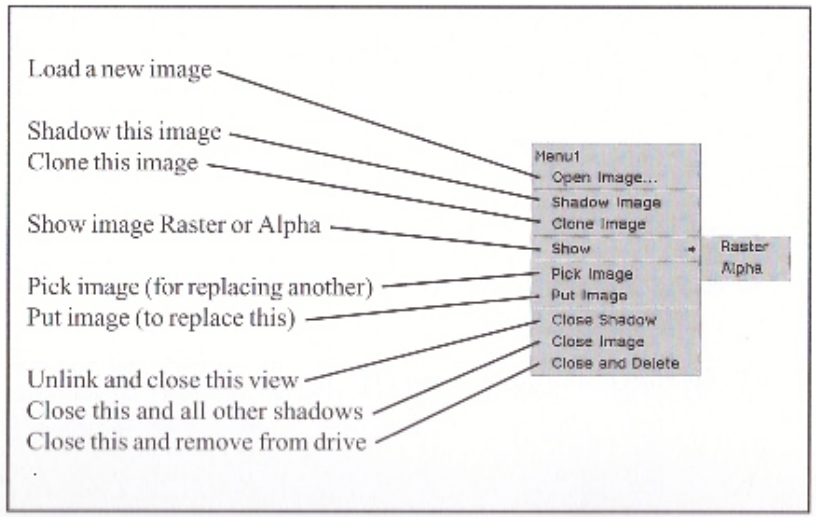
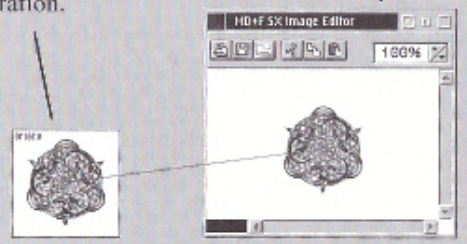
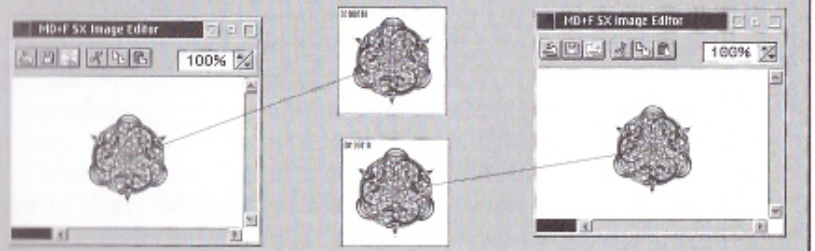


Image data as held internally by WebAK. This data is not affected if you zoom an image or if its colors need to be remapped before being displayed on the monitor due to system configuration.

Image as shown on the WebAK main window, an Editor Window or an Image View Canvas.



There are two views and two separate images after Cloning...



There are two views showing the same image after Shadowing...

indicates a transparent image.

Pick Image - Picking an image is like dragging from the Image View Canvas. Use the Put Image menu item to drop the picked image.

Put Image - Put Image performs the same action as dropping an image onto the canvas. You can use Pick and Put when you are dragging from one source to many targets. In such a case, you can Pick once and Put many times.

Close Shadow - Use this menu item to close this view to the image.

Close Image - Use this menu item to close all views to the image.

Close and Delete - Use this menu item to close all views to an image and to remove (delete) it from the disk

Using an Image View Canvas with Drag-and-Drop

To use the drag and drop feature of Image View Canvases, pick up an image from an Editor Window, an image from an Image View Canvas, or a Bitmaps file and simply drag to the target and drop it there. Make sure you hold down the correct keys for the action you are performing.

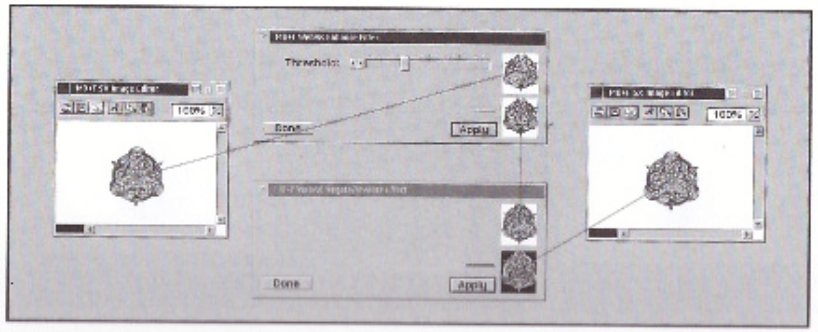
Move and/or Copy

When you move or copy an image, you will be creating a new copy (a clone) of that image in the target Image View Canvas.

Link

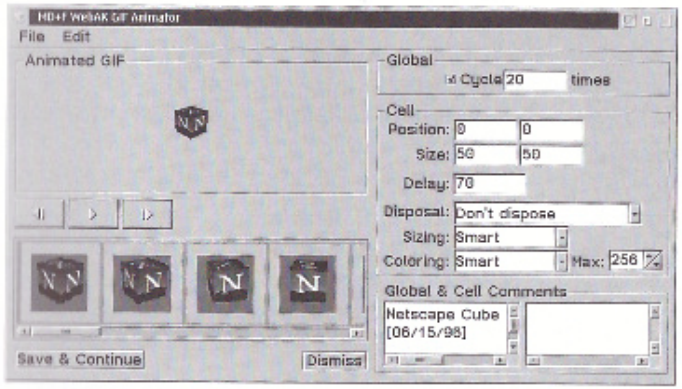
When you link an image, you will be creating another view into that image (shadowing it). Both Image View Canvases, and any others linked, will be displaying the same image data.

You can use drag-and-drop to link the input and output of image operations. For example, you may want to Enhance and then Negate an image. You can shadow your image from the Editor Window into Enhance's source Image View Canvas, drop a copy and into its target and link that with Negate's input Image View Canvas, drop a copy into Negate's target and then shadow it. Now, any changes made to the input image will be enhanced and then negated on the output image using the two linked Operation Windows because of the way you connected their Image View Canvases. In the picture below, the straight lines represent linked images:



Chapter 3: GIF Animator Window

The GIF Animator window displays GIF animations and can be used to load and take apart or examine GIF animations you obtain from the World Wide Web as well as save GIF animations to place on the World Wide Web.



Loading GIF animation files

To load a GIF animation select the **File>Load...** menu item and use the file dialog to open the GIF file. Once the file has been opened, its frames will be displayed on the bottom. Each frame will have its own Image View Canvas so that you can manipulate them individually.

Grabbing and saving a frame

To save an individual frame, simply use the Image View Canvas to either shadow or clone it into a window. Once you do this, you can use the **File>Save As...** menu item on the WebAK main window to save that frame. You can also drag-and-drop the frame onto an Editor Window for saving.

For example, lets say you need to save a copy of the first frame in an animation. To do that, after loading the GIF file:

1 - Move the mouse over to that frame.
2 - Press the right mouse button for the pop-up menu.
3 - Select Shadow Image. WebAK will display your frame in an Editor Window.

4 - Save it just as you would any other image being displayed on an Editor Window.

Adding frames to an animation

To add a frame to your animation, create a blank cell for it by using the frame modification pop-up menu. This menu is accessible from the gray area surrounding each of the frame Image View Canvases. Simply pop-this menu up and insert new animation frames. You can then use the standard Image View Canvas menu items or drag-and-drop your frame's image.

You can add images (or remove them) at any time, even when the animation is being displayed.

Modifying frames

When you load a GIF animation or have created a new one by adding frames, you will normally need to access each frame's settings. To do this, use that frame's pop-up menu (located around its Image View Canvas) and select the **Modify** menu item. The global attributes and that frame's information and settings will be displayed to the right side of the animation window where they can be changed

Each GIF animation has the following global attributes:

Size - A GIF animation file should be large enough to fully contain all the frames it displays. When you start creating a GIF animation, WebAK will start with a very small GIF size and enlarge it as your frames increase in size. This action is performed when you press the **Play** button or when you save the animation. If you wish to change this value or would like to enter a value larger than the area covered by the animation frames, you can do so thru a menu item.

Cycle - Select this item if you wish the GIF to contain a NETSCAPE2.0 extension block. This block will specify how many times the animation is cycled. If you do not specify this option, the GIF animation is to be shown only once.

Cycle Times - Enter the number of times a GIF animation is cycled (restarted from beginning) in this field. A cycle value of 0 means that the animation should be shown once and not cycled and shown again. *Note:* Some Web browsers will stop an animation at either the last or the first frame after a set amount of time has passed even if the cycle times value you specify has not been reached.

Global Comment - This field shows the global comment contained in a GIF animation. Since many programs will loose all but the first 255 characters of your comment or display comments more than 255 characters in

length in varying ways, you may wish to limit it to less than that many characters. This also applies to the comment attached to each frame.

Each animation frame has the following attributes:

Position - These two fields specify the coordinate where a frame is shown. Position (0, 0) specifies the top-left corner of the GIF animation. Generally, your animation's first frame has (0, 0) for its position and (width, height) for its size. The following frames would then specify the corners of the rectangular area where a subframe is to be drawn or the same (0, 0) and (width, height) values if the frame is full sized.

Size - These two fields indicate the size of the frame. You can not change them; they are the same values as the image's width and height.

Delay - The delay field specifies, in milliseconds, the delay time before a frame is drawn. The first time that the GIF's first frame is shown, this value is skipped for that frame. The value you enter will be rounded to the next highest value accepted by the image format. This is 10ms for GIF animations.

Disposal - Disposal indicates the method you wish to use to remove each frame once it has been displayed. The disposal methods are "Unspecified", "Don't Dispose", "Restore Background", and "Restore Previous". Almost all GIF animations use the "Don't Dispose" method since it is the fastest to display and the only one supported by the more popular Web browsers. The method "Don't Dispose" means that the image should not be disposed of. When used, each frame will simply be painted on top of the previous one which means that you will see whatever was shown before thru the transparent areas of your image. "Restore Background" will paint the new image and fill any transparent areas with the background color. "Restore Previous" will show whatever was painted before the frame was drawn. The only exception being the last frame; its generally believed that if a Web browser implements this feature, disposal of the last frame will result in the painting of the first frame in the animation only. Its strongly suggested that you only use the "Don't Dispose" disposal method for GIFs placed on the World Wide Web. Resurrect the collapsed environment file disposal.sxo for three GIF Animators which contain an example of each of the three specified methods.

Sizing - This control lets you specify how each frame is sized when it is saved. You can specify "None", "Auto-Crop" and "Expand-and-Store" Specify "Auto-Crop" to automatically crop the areas of your frame which has not from the previous drawing. The cropping feature does take the disposal setting into account when deciding what extra pixels need to be cropped. "Expand-and-Store" will expand each frame to full size before storing it. You will usually never want to do this unless you need to show GIF animation on a much older GIF animation viewer which requires full

sized frames.

Coloring - This control lets you specify how the image's colors are to be handled. You can use this to specify whether the frame may use the global colormap or the one supplied by the image. This is only an optimization control and will not affect the look of your image.

Max colors - Should an image have more colors that allowed by the GIF specification (256 or 255 colors), it will be remapped to have fewer colors. You can use this control to specify even a fewer number color to achieve smaller file size.

Cell Comment - This field shows the comment attached to the frame you are modifying.

Viewing animations

You can view animations by pressing the VCR style play button. When you do this, the animation will be played frame by frame and will be cycled if you have selected that option. (The cycle option is in the modification area; see above) To stop the animation, press the play button to unlatch it. You can also see the next or previous frame by using the two step buttons. (Do not use the backstep button unless you are using the "Don't Dispose" disposal method for all your frames and you should let the animation cycle once fully after making changes if any frames were added, removed, or modified)

When using WebAK's GIF Animatior window to animate your frames, please take notice that:

1 - The animation speed, specially for very fast animations, may be somewhat different than Netscape's or other Web Browsers.

2 - The frames displayed by the GIF Animator may have more colors in them than allowed by the GIF file format specifications. In addition, on many systems and graphic card configurations, Netscape or other Viewers will use a specific set of colors to display the image, as well as dither it in a user specified way.

3 - WebAK supports more disposal methods than most Web Browsers, including Netscape, support. If you use such disposal methods, your animation may be completely different than you expect when you finally add it to a Web page. Also, a badly made animation obtained from the World Wide Web may display incorrectly if it was specifying an unsupported disposal method and assuming that that setting was being ignored.

Because of the reasons listed above, you should always test your animations with Netscape on a few different computers before using them on a Web page.

Saving GIF animation files

To save an animation you've created, select either the **File>Save** or the **File>Save As...** menu item. When you do this, each of your frames will be readied and saved to disk. Since this step may take some time for animations with many frames, a window will be shown displaying the progress of the save. Because of WebAK's multitasking design, you do not need to wait for the save to finish but you should make sure that you do not modify any of the frames being saved.

After the GIF animation is saved, you may also wish to collapse your environment so you can bring back your GIF Animator Window at a later time without having to setup the frames again in. This is specially important since a saved GIF file may have its frames cropped and optimized (optional) as well as have their colors reduced to the number allowed by GIF specifications when saved. Always keep the originals. For an example of these changes see the changed.sxo file. Resurrect this file and try saving the animation and then loading it in another GIF Animator window to see how it was changed. You will need a 24-bit display to properly observe the changes.

Testing GIF animations with a Web Browser

To test your GIF animation with a browser such as Netscape, either enter the file location in the URL field or create a small HTML file such as:

```
<HTML>
<IMG SRC="C:\TEST.GIF">
</HTML>
```

and open it in the browser's URL field. By using a HTML file, you can try your GIF in front of different backgrounds or show multiple GIFs at the same time.

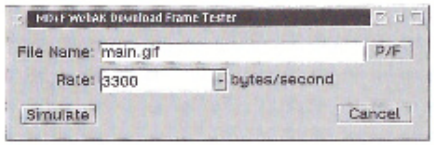
Note: Netscape locks and loads the image on every cycle if you use a file name, if you try to save at the same file location that is being displayed, your save action may fail.

Testing a GIF animation for download behavior

Many Web sites (mostly commercial sites) have started using GIF animation with long playing time for advertisement purposes. These GIFs usually contain hundreds of frames and while you browser a page, display one message after another. When designing such animations, two assumptions are made:

1 - The visitor's browser is capable and configured to display GIF animations as they are being downloaded, and
2 - The download speed is fast enough to load images ahead of the time they are required to be shown. This way, no matter how long the GIF animation is, its frames are loaded just ahead of the time they are displayed. If the animation is too fast or too large for the download speed, it will not be very attractive and if it is loading much faster than being shown, it probably could have been made larger and more interesting.

For such advertisement GIFs (which, in our opinion, are as annoying as GIF animation can get and can be confusing depending on a browser's preference setting) or for situations where smaller GIF animations should properly display while being loaded, you can test them with the Download Frame Tester (DF Tester) tool.



Using the Download Frame Tester

Once you have created a GIF animation which you would like to test:

1 - Save your GIF animation.
2 - Open the Download Frame Tester window using the **Utilites>DF Tester...** menu item.
3 - Enter the GIF name and bytes per character value you would expect most of your viewers will achieve.
4 - Press **OK**.

The DF Tester will then start simulating a download of that speed and measure how long extra waiting time (if any) it took before each frame is shown. This simulation will take only a few seconds and then the results are displayed. You can use the displayed results to reduce the extra delays which may badly affect your GIF animations presentation. The DF Tester will also indicate how much faster or slower than the nominal speed the whole animation loaded and allow you to globally test the GIF by changing all its size or delay values by a percentage.

Chapter 4: Editor Window

Editor Windows are the windows on which you can perform basic editing or modifications to your image. Whenever you open an image file using WebAK or perform a **Clone** or **Shadow** operation on an image, a new Editor Window will be opened and the title of an Editor Window will display the name of the image it displays. (if it has a name)

The Editor Window's toolbar, shown in the above picture is optional, To remove it select **Options>General...** for the general options window and set its setting to **None**.

The editing features provided by an Editor Window are:

- Changing an image data format such as bits per pixel.
- Creation of an image raster or an Alpha channel from the other.
- Resizing of the image.
- Cropping of the image.
- Rotating it.

as well as saving it in a different format.

Opening Editor Windows

To open an image in an Editor Window, select the **File>Open...** menu item and choose the file to be loaded in the file dialog. WebAK will attempt to load and display the file and will either show you your image or display a large X. A large X is an indication that there was a problem with your image file.

Note: Since most common file formats (such as GIF, JPEG, IFF/ILBM, or PNG) are supported, a large X will usually indicate a damaged file.

Once the file is loaded or Xed out, you can create additional views into it by using the window's pop-up menu and selecting **Shadow Image** for additional Editor Windows displaying the same image data.

Viewing at different zoom levels

To view the image at a different zoom level, enter the percentage value into the field in the Editor Window's tool bar or use one of the menu items to modify that value. Enlarging the zoom level is specially useful when working with small images commonly used on World Wide Web pages.

Changing image or Alpha channel format

An image's data can be in one of the following three formats:

- 8 bit indexed (with a color palette)
- 24 bit true-color
- 48 bit precision true-color

In addition, the 8 bit indexed format can be created to contain only black and white or grayscale colors. You can switch between any of these formats by using the **Palette>Convert To** menu items.

The image's Alpha channel can also be contained in two different image formats:

- 8 bit (256 levels)
- 16 bit (65536 levels)

To change from one format to another, use the **Image>Alpha>Convert To** menu items.

Creating image or Alpha channel data

There are times when it would be useful to create an Alpha from the image data. (or vice versa) To do that, use the **Image>Alpha>Generate** menu items. Normally one would use this technique with images from painting or drawing programs which do not support direct manipulation of an image's Alpha channel.

In addition to the ability to create an Alpha channel from the image raster data, you can create full Alpha channels (fully opaque images) as well as empty Alpha channels (fully transparent). Use the **Image>Alpha>Generate>Full** or **Image>Alpha>Generate>Empty** menu items

for that. By default, unless an image file contains Alpha channel information, the image will be assumed to contained a full Alpha.

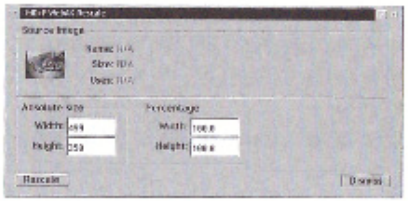
Changing image size

To resize an image or resize the canvas of an image, use the Rescale window. This window lets you create new images which are smaller or larger than the current image or you can select to keep the image the same size but expand or shrink it's canvas area.

Resizing an image

To resize an image, enter its width or height, either in pixels or in percentage change, in the **Width** and **Height** fields and then press **Rescale**. A new editor window will open and display a clone of your image with the dimensions you specified.

Resizing a canvas



To resize the canvas your image is drawn on without resizing the image itself, enter the width and height of the desired canvas, select the expand option, and then press **Rescale**. A new editor window will open and display a clone of your image on a canvas of the specified size. When resizing a canvas, you can either keep the image centered or align it to one of the sides or corners of the new canvas. A centered image is the default but to do an alignment, press the Alignment button until the desired alignment is shown. If you do this and follow it by pressing **Rescale**, the image canvas will be expanded or shrunk (and clipped) in the opposite direction of your selection.

Making precise size adjustments

The Rescale window contains two sets of entry fields for you to specify or modify your images dimensions:

1 - The **Width** and **Height absolute size** fields which indicate the resulting image's width and height.
2 - The **Width** and **Height percentage** fields which indicate the resulting image's width and height as a percentage of the source image's width and height respectively.

You can however use these fields to specify the desired dimensions in a few other ways:

Absolute size change - To specify change in dimension, enter the amount of change, such as +10 or -10 into the absolute size fields. For example, to add 20 in each direction to a 1024 by 768 pixels image, you can enter +20 into the **Width** and **Height** fields. WebAK will then calculate and display 1044 and 788 in those fields and also update the percentage fields.

Percentage change - To specify change in percentage, enter the amount of the change, such as +25% or -25% into the absolute size fields. For example, to make a 1024 by 768 image 25% smaller in each direction, you can enter -25 into the **Width** and **Height** fields. WebAK will then calculate and display 75% in each of those fields (assuming they were 100%) and also update the absolute size fields.

Cropping the image

To crop an image, you must select a rectangular area of it and then select the **Tools>Selection>Crop** menu item. To select a rectangular area, you should be in the arrow mode which you can access thru **Tools>Arrow**. Once you have selected an area, you can resize a pixel at a time by clicking close to its edges or remove the selection by using **Tools>Selection>Kill**. You should not leave a selection on a window unless you intend to use the selection.

Rotating or mirroring the image

For most image processing operations you will need to use an Operations Window, however, since rotating an image in 90 degree angles or mirroring it is a common operation, you can perform them directly within an Editor Window. Use the rotate or mirror menu items at the **Image** menubar to perform these actions.

Saving the image

Once you have modified an image, you can save it by selecting the **File>Save** or **File>SaveAs...** menu items. The image will be saved using the same file format that it was loaded in unless you change it.

File name extension

The WebAK application ignores an image files extension and checks the actual image data to derive at its file format but there are many popular applications do not do that. It is recommended that you use the standard "filename.ext" format without any unusual extensions. In addition, you should take caution at using more than one "." or the characters space, "#",

"?", "/", "*", "~", "[", "]", "!", "@", "(", ")", ":", ";", " or "\" . The problems you may encounter with these will not only depend on the operating system which you are using while accessing one of these images but also on your individual applications. For example, many programs written by IBM have trouble with extra periods in a file name and many UNIX programs have trouble with ";". Again, it is recommended that you use the standard "filename.ext" format without any special characters.

Saving images, WebAK style

One of WebAK's unique features is its ability to Collapse and Resurrect environments (discussed later in this manual) By using the Collapse and Resurrect features, you can save your image while it is still in its Editor Window or in an Image View Canvas located on an Image Holder or an Operations Window. You can also store a large number of images in one step by using Collapse and Resurrect. You may wish to use that feature for better management of images, specially images which you are using in your current work and will not be storing for the long term.

You will see examples of this in many of the collapsed environment files supplied by MD+F. In many cases, those files contain image data and may even contain the name of those image files (such as image_back1.gif") but are not shipped with the individual image files.

Since many of the objects as well as the Collapse and Resurrect features talked about in this section are covered by other parts of this manual, you need to wait until you have read those sections before deciding what is the best way for your short-term image storage; however, please try resurrecting these three collapsed environment files for a quick example:

- editimg.sxo - Three image stored in Editor Windows.
- holdimg.sxo - Three images stored in an Image Holder.
- chanimg.sxo - Three images stored in the Operations Window which created them to begin with.

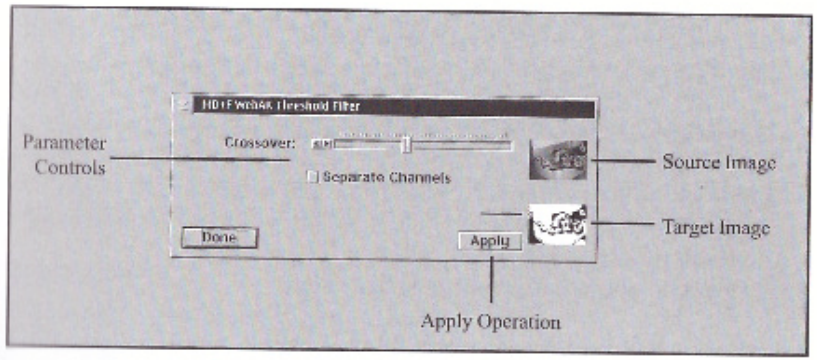
and if you wish to load the images the old fashioned way:

- red.gif, green.gif, blue.gif - The three images in individual files.

Chapter 4: Operations Windows

MD+F WebAK provides a number of image operation (filters and effects) windows. These windows can perform simple operations such as negating an image or as complex as merging two images based on the intensity of a third one. Fortunately, the interface to these Operations Windows is very consistent from one Operations Window to the next and once you have mastered how an Image View Canvas works, you should be able to use them without much trouble.

A typical operations window



has the following component:

Parameter Controls - This area holds a number of controls which effect how the operation is performed. The controls effect parameters such as the intensity of the operation being performed.

Source Image View Canvas - This image view canvas contains the source image used in the image processing operation.

Target Image View Canvas - This image view canvas contains the results of the image operation once it has been performed. You can either save your results or link this Image View Canvas to another Operation Window's source Image View Canvas to link the effects.

Apply - This button needs to be pressed for the operation to be performed. Note that there are some exceptions to this, specially when operations are linked.

Done - Pressing this button will save the values in the Operation Window's controls and closes its window. The values contained in the window will be used the next time you open an Operation Window for the same filter or effect.

Opening an Operations Window

To open an Operations Window, select the operation you would like to perform from the menu items contained in the **Image** menubar. Once you do that, an Operations Window of that type will be opened. For example, if you select the **Image>Brightness** menu item, a Brightness operations window will open.

Using an Operations Window

The steps for using a typical Operations Window are as followed:

1 - **Specifying a source image**. You will need to either drag-and-drop your source image to the source Image View Canvas or use that Image View Canvas' pop-up menu to open an image into it.
2 - **Specify a target image**. This is the image which will be overwritten (or partially overwritten in some cases) An easy way to create it is by drag-and-dropping a copy of the image contained in the source Image View Canvas into it.
3 - **Enter operation parameters**. Modify the parameter controls to your liking. If any of the parameter controls are Image View Canvases, drop your control images into them.
4 - **Perform the operation**. Press the **Apply** button.

Now, you can drag-and-drop the results of the operation from the target Image View Canvas to wherever you wish.

Resurrect operwin1.sxo for an example of this. Notice how the two editor windows are linked to the Image View Canvases of the Operations Windows and how by changing the Operations Window's controls, you effect the resulting image in its target Image View and the corresponding Editor Window.

Linking Operations Window

Linking Operations Windows allows you to perform more than one operation on an image. The usual way to do this is to setup any number of Operations Windows as instructed above and then link target to source Image View Canvases between the operations. Then, the first operations window linked contains the unlinked source and the last Operations Window linked contains the unlinked target.

Resurrect operwin2.sxo for an example of linked operations window. Try changing the parameter settings on the Operations Windows and pressing

the **Apply** buttons to see how things they are linked.

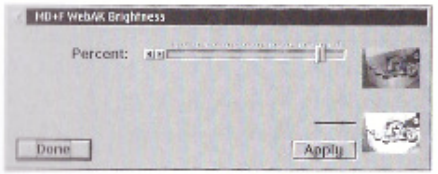
When linking Image View Canvases, you should never create conditions where there are loops in between your windows or in-between the Image View Canvases of the same window. Such loops are not supported by this product. If you decide that such a loop is necessary and useful in a large and complex linked operations setup, you may use them but you should be aware of the risks as demonstrated above as well as change of behavior in future versions of the product.

Once you've mastered using Operations Windows and linked Operations Windows, try the interesting sol.sxo collapsed environment file; notice how you can expand on it by using the color channel operations or by inserting additional filtering operations at various points.

Operations (Filters and Effects) provided by WebAK

Since Modular Dreams Incorporated continuously seeks to improve its products, the operations dialogs shown in this section may be slightly different than those running on your machine and there might also be more of them as added by future upgrades. However, the behavior and use of the actual operations should be similar to what is discussed in this section.

Brightness



Adjust brightness of image.

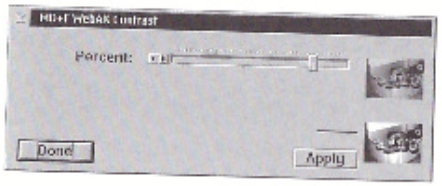
Usage:
1 - Provide images in the source and target Image View Canvases.
2 - Specify how much modification is desired using the **Percent** slider.
3 - Press **Apply**.

Combine Color Channels

Create an image by merging images which specify its separate color channels.

Usage:
1 - Provide images in the source and target Image View Canvases.
2 - Press **Apply**. (Any unnecessary information in the source images will be ignored.)

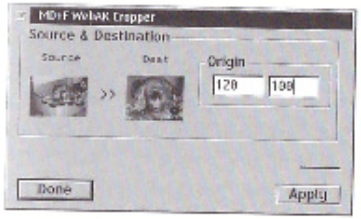
Contrast



Modify contrast of the image.

Usage:
1 - Provide images in the source and target Image View Canvases.
2 - Specify how much change is desired using the **Percent** slider.
3 - Press **Apply**.

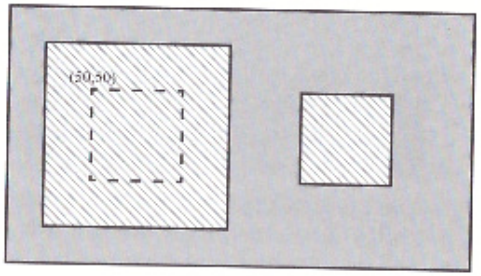
Cropper



Crop or stamp an image. This operation allows you to cut out a portion of an image or to stamp an image onto a larger one.

To use for cropping a larger image into a smaller one:

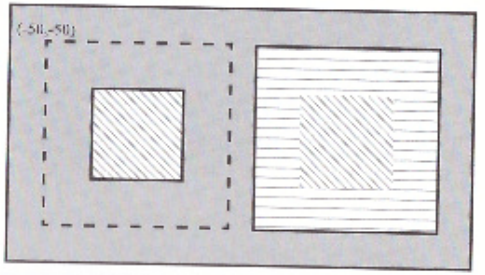
1 - Provide the source and target images into the proper Image View Canvases.
2 - Enter the **Offset** values. The offset values indicate at what offset from the source image should the target be copied from. For example, if you provide a source which is 200 by 200 pixels and a target which is 100 by 100, offset of values of 50 and 50 would copy the center 100 by 100 pixels of the source are copied into the target.



3 - Press **Apply** to perform the action.

To use for stamping a smaller image into a larger one:

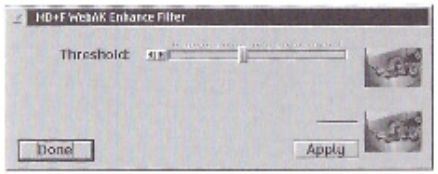
1 - Provide the source and target images into the proper Image View Canvases.
2 - Enter the **Offset** values. The offset values indicate at what offset from the source image should the target be copied from. For example, if you provide a source which is 100 by 100 pixels and a target which is 200 by 200, offset of values of -50 and -50 would copy the source image into the center 100 by 100 pixels of the target.



3 - Press **Apply** to perform the action.

Please note that this operation does not perform two different tasks. It is

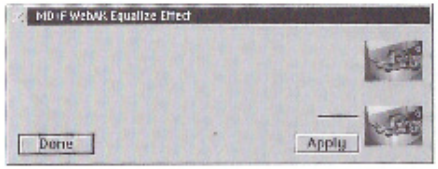
Enhance Filter



Enhance image quality, specially around edges.

Usage:
1 - Provide images in the source and target Image View Canvases.
2 - Select the **Threshold** value used to decide which pixels are enhanced.
3 - Press **Apply**.

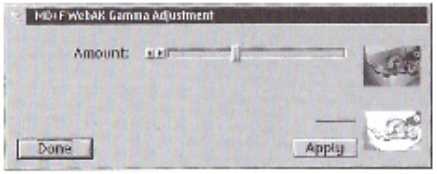
Equalize Effect



Equalize image range. An equalized image is the same image expanded to fill the full brightness spectrum. Normally, you will only want to apply this filter to an image which should contain both colors White and Black.

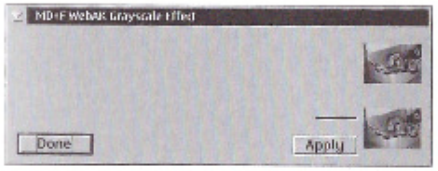
Usage:
1 - Provide images in the source and target Image View Canvases.
2 - Press **Apply**.

Gamma Adjustment



Usage:
1 - Provide images in the source and target Image View Canvases.
2 - Select the adjustment **Amount**.
3 - Press **Apply**.

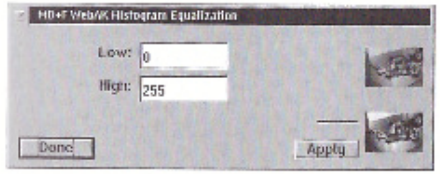
Grayscale Effect



Make the image grayscale by combining the Red, Green, and Blue components into one.

Usage:
1 - Provide images in the source and target Image View Canvases.
2 - Press **Apply**.

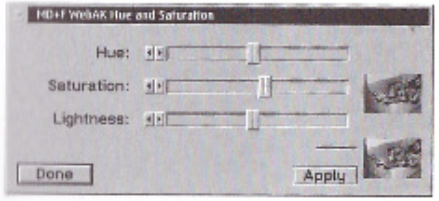
Histogram Equalization



Equalize the image histogram and expand it for the range provided

Usage:
1 - Provide images in the source and target Image View Canvases.
2 - Enter the minimum and maximum values.
3 - Press **Apply**.

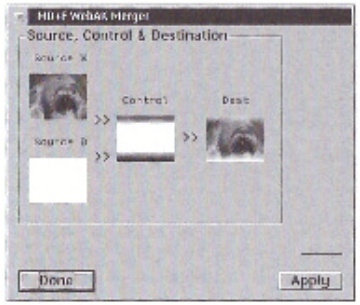
Hue And Saturation



Modify image HSL values.

Usage:
1 - Provide images in the source and target Image View Canvases.
2 - Increase or decrease the HSL values by moving the sliders away from the center of their range.
3 - Press **Apply**.

Merger



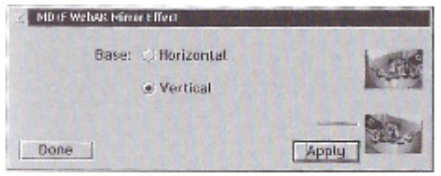
Merge two images into one based on a control image.

Usage:

1 - Provide two images in the source Image View Canvases. One of the images will be used for darker pixels and the other for lighter ones.
2 - Provide an image in the target Image View Canvas.
3 - Provide an image in the **Control** Image View Canvas. This image controls how the two source images are mixed. A value of 0 (black) means use a pixel from **Source B** and a value of 255 (white) means use a pixel from **Source W**. Values between white and black will result in a mixture of the pixels in **Source B** and **Source W**.
4 - Press **Apply**.

For example uses of this Operations Window, resurrect the following collapsed environment files: merge1.sxo, merge2.sxo, merge3.sxo, merge4.sxo, or merge5.sxo. In each case, once the file has been resurrected, either press the **Apply** buttons (if dialogs are opened) or press the **Execute** button. (only if a script window is opened)

Mirror Effect

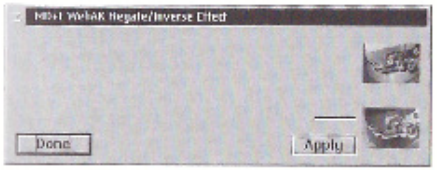


Flip image in the horizontal or vertical direction.

Usage:

1 - Provide images in the source and target Image View Canvases.
2 - Select **Horizontal** or **Vertical** mirroring.
3 - Press **Apply**.

Negate/Inverse Effect

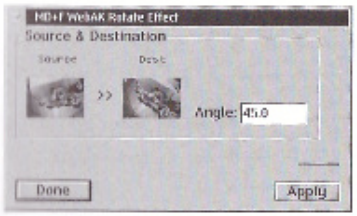


Invert image colormap.

Usage:

1 - Provide images in the source and target Image View Canvases.
2 - Press **Apply**.

Rotate Effect

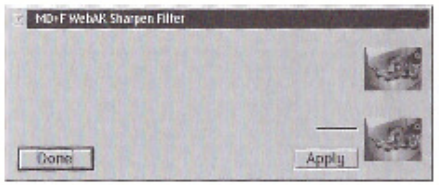


Rotate an image in the clockwise or counterclockwise direction.

Usage:

1 - Provide images in the source and target Image View Canvases.
2 - Enter a value between -360 to 360 for rotation **Angle**.
3 - Press **Apply**.

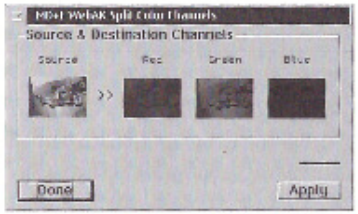
Sharpness Filter



Sharpen image.

Usage:
1 - Provide images in the source and target Image View Canvases.
2 - Press **Apply**.

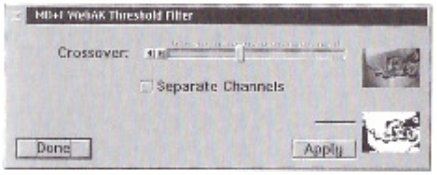
Split Color Channels



Create images by splitting an image into its separate Red, Green, and Blue color channels.

Usage:
1 - Provide images in the source and target Image View Canvases.
2 - Press **Apply**. (Any unnecessary information in the source images is ignored.)

Threshold

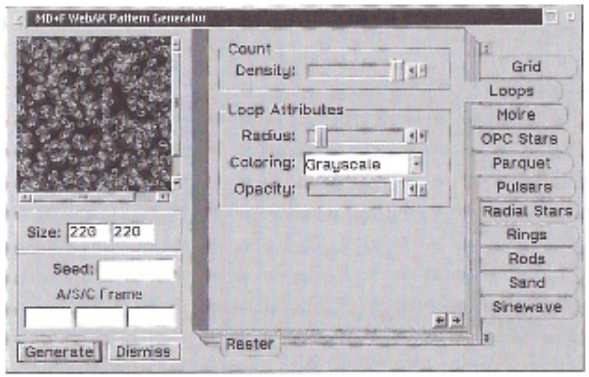


Create a Black and White image based on the source images color intensity and a cutoff value.

Usage:
1 - Provide images in the source and target Image View Canvases.
2 - Provide the **Crossover** value which indicates the pixel brightness which is the crossover from Black to White.
3 - Select **Separate Channels** if you would like WebAK to perform three separate threshold operations on each of the Red, Green, and Blue color channels.
4 - Press **Apply**.

Chapter 7: Pattern Generator

One of the many World Wide Web related features provided by MD+F WebAK is the Pattern Generator. The Pattern Generator is capable of creating seamless patterns for use as generated or as the starting block for you to create your own custom patterns.



Unlike many other windows in this application, you can have only one Pattern Generator window open at one time and it may not be collapsed. The normal way to store the patterns you create is using the Image Holder object.

Selecting pattern size

You can entered the desired pattern dimensions in the two entry fields labeled **Size**. Some patterns may have special requirements for their sizes - for example, they might be required to be even - so the size you enter might be adjusted by the application before generation of the pattern.

Using an image seed

If you would like, you can enter a random generator seed value for WebAK to use. The seed value will be used to set the random generator to a specific setting so that the sequence of the numbers generated are known. Using an **Image Seed** will allow you generate the same pattern later, assuming you also keep the other pattern generator settings the same.

Normally, this field is blank and a random seed will be used for each **Generate** operation.

Using animation frames

Since some of the pattern styles are ideal for creating animations, there are three fields provided to allow you to specify the **Angle**, **Speed**, and **Count** of the objects in some of the pattern styles. When used with any given image seed, the Pattern Generator will perform the following actions:

1 - Create all the random values it needs.
2 - Modify the random values based on the animation frame values you've provided.
3 - Render the final image.

The actions these three settings perform are as followed:

Angle - Specifies the angle rotation frame. The larger this number is, the more the objects in the animation frame have rotated. This value is in degrees. Any values above 360 are mapped to their corresponding value between 0 and 359.
Speed - Specifies the speed frame (effects location). The larger this number is, the more the objects have moved from their starting position. This value does not have a limit.
Count - Specifies the count frame. The larger this value is, the more of the items are drawn. This number will range from 1 to the maximum number of items specified by the pattern's density value.

When using the frame values, please notice that they are at they advance at the slowest possible increments. For faster animations, you should simply skip over the in-between frames.

For sample animations created using these fields, resurrect the following collapsed environments: pattern1.sxo, pattern2.sxo, and pattern3.sxo.

Generating a pattern

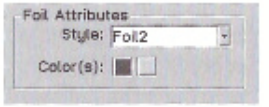
To generate a pattern, enter its size, select the pattern type from the notebook, set its settings (if any), and then press the **Generate** button. As with all other operations in the multitasking MD+F WebAK, the creation will be started as a separate task and you may continue using the rest of the application while the pattern is being created and displayed.

Pattern styles

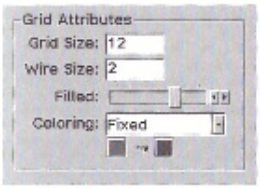
Since Modular Dreams Incorporated continuously seeks to improve its products, the patterns style pages shown in this section may be slightly different than those running on your machine and there might also be more of

them in your software. However, the behavior and use of the actual style pages should be similar to what is discussed in this section.

Foil



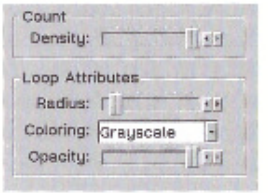
The Foil module creates classic foil paper wrap patterns based on the foil style selected thru the **Styles** selection list. The size of the Foil patterns generated depend on the image canvas size. The only options available for these styles is the color(s) of the foil and the background, if any.



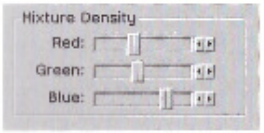
Grid

The Grid module creates a wire grid where each cell of the grid is either empty or contains a ball. The width of the grids is specified by the **Wire Size** and the size of each of the cells is specified by **Grid Size**. You can select what percentage of the grids contain balls by using the **Filled** scrollbar and select the coloring scheme used to paint the balls through the **Coloring** list. If you would like to use specific color(s) instead of the colors in the predefined coloring schemes, you use the color option(s) to specify the desired colors and choose the **Fixed** coloring scheme.

Loops

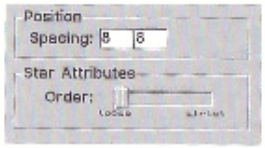


The Loops module draws a number of loops (spirals) on the image canvas. You can specify the density of the drawing by using the **Density** control. You will have very few loops if you select a low density and many if you select a high one. **Radius** controls how large each loop is. The loops will be colored according to the coloring scheme you select in the **Coloring** list and the opacity of each loop is selectable through the **Opacity** scrollbar. The less opaque each loop is, the easier it will be to see through it.



Moire

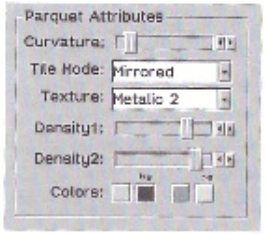
The Moire module will draw a simple moire pattern consisting of red, green, and blue components. The size of the image and the density you specify effect the results.



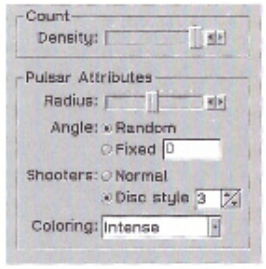
OPC Stars

The OPC Stars module paints a blank sky and stars similar to those used by the Orion Picture Company in their opening animation. You specify the average spacing of the stars in the horizontal and vertical directions using the **Spacing** fields. The **Ordering** selects how loose or ordered the stars are from their specified locations. A fully ordered image will place the stars at equal distances from each other and a fully loose ordering will allow them to be placed randomly at any point in the area specified by **Spacing**. By specifying a minimum ordering value of 10% or so, you can assure that the stars are never placed too close to each other and never touch one and another. The color of the stars and the background are specified using the two color buttons.

Parquet



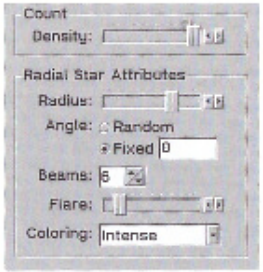
The Parquet module creates a tiled pattern by dividing the image canvas into four sections of equal areas and painting the tiles in pairs. There are three tile **Textures**: **Normal**, **Metallic 1**, and **Metallic 2**. The **Normal** texture consists of lines drawn parallel to each other and these lines may be straight or curved, as set by the **Curvature** control. The other two textures, **Metallic 1** and **Metallic 2** draw pressed metal textures. Using **Tile Mode**, you can specify how the files are arranged against each other, for example, you can have tiles opposite of each other to be mirror imaged or placed the same way. For all three tile textures, you can specify the density of the drawings. There are two density controls, one for the upper-left and lower-right tiles and the other for the upper-right and lower-left ones. There are also two color buttons for the two pair of tiles.



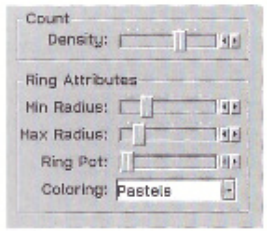
Pulsars

The Pulsars module draws pulsar or disk shaped objects. You can specify the density of the drawing using the **Density** button. A higher density means a larger number of pulsars are to be painted and a lower setting means fewer are to be painted. The **Radius** setting controls the size of the pulsars, which are all of the same size. They can be drawn at various angles or all be drawn using a fixed angle and may contain the standard two shooters or a different number of shooters (**Disc Style**). The final setting for pulsars is the **Coloring** scheme which is used when the random pulsar colors are being generated.

Radial Stars

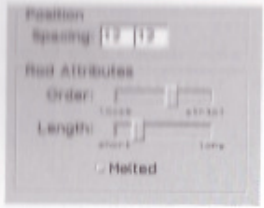


The Radial Stars module is similar to the Pulsars module. The **Density** and **Radius** behave the same way in selecting the number of radial stars and their size and like the pulsars, you may select the radial stars to be drawn at **Fixed** or **Random** angles. The **Beams** control selects how many light beams are drawn and **Flare** indicates their thickness and spread. A high flare value will make the radial stars' beams spread from around the center (similar to a star fish) and a very low flare value will generate very bright and sharp beams. The **Coloring** control select the coloring scheme used when generating the random radial star colors.



Rings

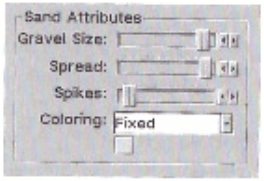
The Rings module paints rings or tubes. The **Density** control specifies how many rings there are and the **Min Radius** and **Max Radius** limit the range of radii used when generating the rings. The amount of the circle actually drawn into the ring is specified by the **Ring Pct** control. The higher you make this value is, the smaller the hole inside of the ring will become. The **Coloring** control selects the coloring scheme used to generate the random colors used in painting of the rings.



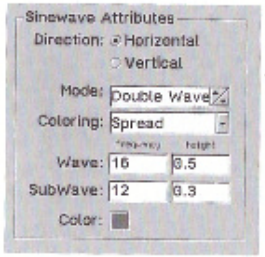
Rods

The Rods module paints Neon colored rods. Specify the spacing and strictness of the rod placements using the two Spacing and Strictness fields and the length of the rods using the Length entry field. To paint White melted rods, select the Melted button.

Sand

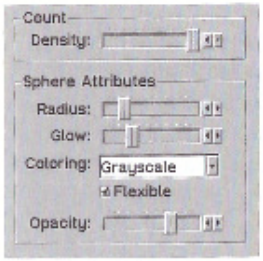


The Sand module will paint a sand or gravel like pattern, usually seen as background on web pages. There are three controls used to specify the shape, size, and roughness of the pattern and one pair of color controls to specify a coloring scheme or a single fixed color. If using a single color, the sand or gravel items are drawn using the specified color and the background color will be black.



Sinewave

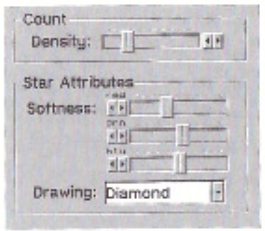
The Sinewave modules draws a Sinewave with a variety of coloring options. You can specify a **Horizontal** or **Vertical** wave using the **Direction** radio buttons. You can also select for there to be another wave added your wave by using the **Mode** control. The **Frequency** and **Height** fields specify the wave attributes for the main wave and the subwave, if any. If you would like only one full wave per image canvas width, specify one, else, enter the number of peeks as the **Frequency**. Similarly, if you like the wave's height to cover the full image canvas height, use a **Height** value of one. A smaller height value will give you a much smoother wave and a higher value will make the wave much sharper. The **Color** button specifies the color using while shading the wave and the **Coloring** control specifies how you would like the wave to be colored.



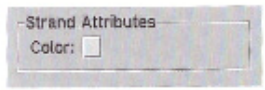
Spheres

Use the Spheres module to draw colored balls or orbs. Similar to other modules like Pulsars or Loops, use the **Density** and **Radius** controls to specify how many spheres there are and their radius. The **Glow** setting is similar to **Radius**. It specifies the radius of the glow within the sphere. If you specify a glow value close to the sphere's radius, you will obtain normally shaded spheres and a smaller glow value will result in glowing orbs. A larger glow value than the radius will generate high-contrast spheres. When painting the objects, the Spheres module uses the **Flexible** setting to decide whether spheres redraw over each other or merge together and the **Opacity** setting to draw partially transparent spheres. Finally, the coloring scheme used is specified by the **Coloring** control.

Stars

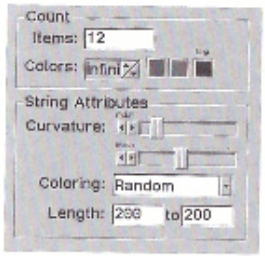


The Stars module will paint dense, cartoon like star fields. Use the **Density** control to indicate how dense the star fields are to be drawn. The **Drawing** control allows you to select **Diamond** or **Cross** shaped stars. The coloring of the stars are is controls by the three **Softness** settings. Each of the softness bars specifies how little variation can there be from pure white for each of the colors. If you set all three fields to maximum, you will have all white stars and if you set one or more of the **Red**, **Green**, or **Blue** softness controls to a lower value, you will have stars randomly colored with the color raging from white to those color(s).



Strands

The **Strands** module paints slightly waving strands in the vertical direction. The only control for this module is the **Color** button which specifies the color used to paint the individual strands.



Strings

The **Strings** module paints strings of varying length and smoothness. Use the **Items** field to specify the exact number of strings drawn. The two **Length** fields indicate the minimum and maximum string lengths the program will paint. Similarly, the two **Curvature** controls indicate the minimum and maximum amount of curvature allowed. By using these four fields, you can modify the look of the strings to mimic objects such as hair (almost identical curvature min and max, one color) or fibers such as those seen in recycled paper. (low curvature and short lengths, two dark colors) The **Color** control specifies if you would like one color, two colors, or an infinite number of colors. If you decide on an infinite number of colors, use the **Coloring** control to indicate the coloring scheme, else, use the color buttons to indicate the fibre colors. Also, use the background color button to indicate the background color.

Weave

The Weave modules will draw a pair of 3D woven ribbons. The size of the image canvas indicates the density of the weaving as there is only one ribbon painted in each direction. The only two options are the two **Colors** color buttons which specify the ribbon colors.

Special uses for patterns

Although the obvious use for the seamless patterns generated by MD+F WebAK is creation of webpage backgrounds, there are also great for other special purposes:

Creation of displacement maps

If you have a paint program which can displace images based on a control image's brightness, you can use patterns for animated patterns to create images whose intensity can be used to displace an image in very interesting ways. Resurrect the collapsed environment file dispmap.sxo for an example of this.

Creation of merging maps

Most paint programs allow merging of two images based on a control merge map. Similar to the displacement map mentioned above, the patterns created by WebAK can be used to control merging of two images. Now, instead of using a pattern's intensity to dissolve an image, you can fade from an image to another using the patterns. Resurrect the collapsed environment file

mergmap.sxo for an example of this.

For a detailed description of the method and applications used for the two examples given please visit Modular Dreams Inc.'s Web site at www.modulardreams.com. We will attempt to periodically add additional examples to out Web pages.

Chapter 7: Image Holder

The image Holder is the simplest tool provided by MD+F WebAK. It is simply used to hold a number of Image View Canvases. It can be used to hold images which you are working with and would like to have accessable without providing for the extra window space taken by Editor Windows or it can be used to store a large number of images in one file. An example use of Image Holders is the collapsed environment file sines.sxo. When we created these patterns for Web design use, we could have stored them one by one but we decided to keep them together in a pair of Image Holders. Whenever needed, they can be loaded in one bunch by selecting **File>Resurrect...** menu item and entering the collapsed environment's file name.

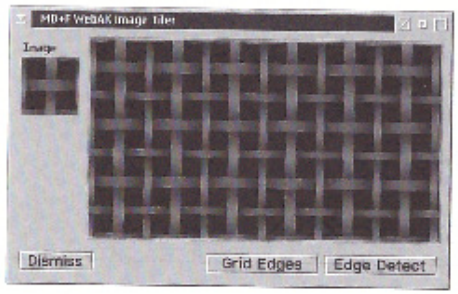
To use an Image Holder, just use its individual Image View Canvases as you would normally.

Changing number of Image Holder items

You can change the size of an Image Holder to increase or decrease the number of Image View Canvases it holds. You need not worry about making it too small, if any of your used Image View Canvases are removed, the Image Holder will maintain its image and redisplay them when enlarged again.

Chapter 8 Image Tiler

Many of the patterns created by the Pattern Generator are seamless since they are intended to be used in creation of Web page backgrounds. The Image Tiler is a simple tool which will tile an image for you. This way you can clearly see how well your image tiles.



To use the Image Tiler, specify your source image in the **Tile** Image View Canvas and it will be automatically tiled in the display portion of the window.

Note: You can link to the **Image** Image View Canvas and have the tiled portion of the window get updated automatically as the image changes.

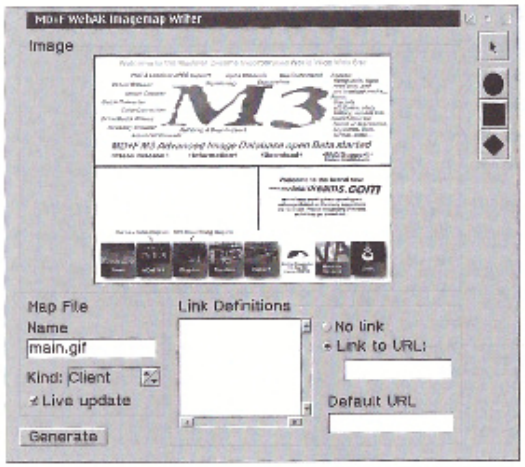
Looking for trouble spots

In order to assist you in detecting any problems with your tiles, you can configure this tool to apply an edge detection algorithm to the image before its shown. You can configure this feature to be applied to the whole image or just a part of it. This feature will assist you in seeing any tiling problems by highlighting areas surround the rectangular tiles you have created.

If you need to find out where the edges of the tiles are, press the **Show Edges** button and they will be highlighted for a short period of time.

Chapter 9: Imagemap Writer

Imagemaps provide a way for a Web page designer to place links to other sites or URL accessible files from an image. MD+F WebAK's Imagemap Writer allows you to load in an image, select areas on it and provide the corresponding links and then generate the appropriate style of an imagemap file from the provided data.



Providing the image

The first thing you need to do in order to create an Imagemap data file using this tool is to provide it the image you are creating a map for. To do this, drag-and-drop or load the image into the **Image** Image View Canvas. Once you do this, the image will also be displayed in the larger working area.

Painting the link areas

On an imagemap, areas which are linked are selected thru basic geometric shaped. The WebAK program supports the shapes Rectangle, Circle, and Polygon. To create a link of any type, press the button containing that shape and then draw within the image area. For points which are contained by more than one shape, the one that comes first is the one which whose link is followed by the Web browser or server.

Specifying links

Once you have drawn your shape, it will be added to the list at the bottom of the window. You can then enter the link location for that area. Should you wish to modify the links for any other area, select it from the list and change the link value shown for it.

Each area can have one of the two possible link values:

- A URL link
- A no link setting. Select this if you want the area to not link to anywhere. You will normally do this to exclude an area from being processed by the default link. The default link is the one that covers all the areas not covered by any of the shapes you have drawn.

Modifying entries

To modify an entry, simply select it and either redraw it on the image (if you wish to change its shape or coverage) or re-enter the link value (this changes its link information) You can also delete the entry you have selected by pressing the **Delete** button.

Selecting the map file format

An imagemaps mapping data can be either contained in the HTML Web page which is loaded onto the user's machine (client side) or be resident on the server machine (server side). You should specify the type of the imagemap data you want created by using the **Kind** selection control. This list also contains a third entry to create a test file. You can use that to generate a complete HTML file and test it with a browser running locally.

Note: Some Web Browsers, such as IBM's WebExplorer, do not support client side Imagemaps.

Generating a map file

Once you have finished specifying the links from the image, enter the file name into the **File Name** field and press **Generate**. This will write the map file out for you to use. You can also create map files on the fly; to do this, select the **Live Update** option. Now, any changes you make to your links or their areas will result in updates to the map file as they happen.

Chapter 10: Collapse and Resurrect

Collapsed environment files (SXO) are Meta-format files used by MD+F WebAK for reading and writing groups of objects and their relationships onto disk. They provide easy means for you to store or share with others the setups you create in order to put together complex GIF animations.

In order to understand the role of collapsed environment files in WebAK, its best to compare them with traditional storage files:

All applications, including WebAK allow for storage and retrieval of various file formats. In WebAK for example, you can store your images using standard graphic file formats (.gif, .jpg, ...), store your scripts in standard ASCII format (.sxs or .txt), etc. The reason why all applications provide support for these standard file formats is so that the you can save and retrieve the same files by different applications; for example creating a graphic file using one paint program and then later modifying it on another.

In addition to standard file formats, WebAK provides the additional file format SXO in order to allow you to store and retrieve objects of varying complexity with ease. For example, if you open an Operations Window and modify its controls, you can collapse the environment (using **File>Objects>Collapse**) and recreate the object in the future by resurrecting it. In such a case, there could have been another file format used: one that simply saves the setting of the controls. Now imagine two such Operations Windows with the target of one linked to the source of the other. Now, using a traditional settings file, you could store the setting for one, store the setting for the other but what about the rest of the environment? In order to get back to where you where, you would need to reopen the Operations Windows (i.e. remember which two they were), load their settings into them, and link the source and target canvases. (i.e. another thing to remember) To do this the WebAK way, you would have simply collapsed the environment to save everything and the next time you needed the same setup, resurrected it to bring everything back. And the difficulty stays the same even as the setup get more complex. (as opposed to traditional load and save methods which would require more files and more steps to follow)

Details of SXO

The SXO file format stores all MD+F WebAK windows which are used in creation of GIF animations. This includes Editor Windows, Operations

Windows, Image Holders, GIF Animator, ... What is not stored are application windows and dialogs such as the various application settings windows or file dialogs. The only exception is the Pattern Generator window which can not be collapsed.

File versions

When SXO files are read or written, WebAK will skip over information it does not understand and make educated guesses for information which it needs but can not find. This allows WebAK to load (or load as much of) SXO files created by earlier or later versions of the software and also allows Modular Dreams Inc. to store more environment settings in the file or modify the some of the windows with minimal chance of making older SXO files unusable. The WebAK application does not provide file version information when an environment is resurrected and will automatically save a files in the latest version when they are collapsed.

Sharing SXO files

To share an SXO file with another WebAK user, you need to provide them the following items:

- The SXO file.
- The images used by the file's objects only if you read the images using WebAK scripting language.

and you need to make sure that the windows in your collapsed environment file are not badly placed. For example, do not use a full 1600 by 1200 display are if the person you are providing the file to has a 1024 by 768 display.

Note: SXO files compress well; You may want to use a tool such as Zip to compress them if you will be sharing them with other thru the use of diskettes.

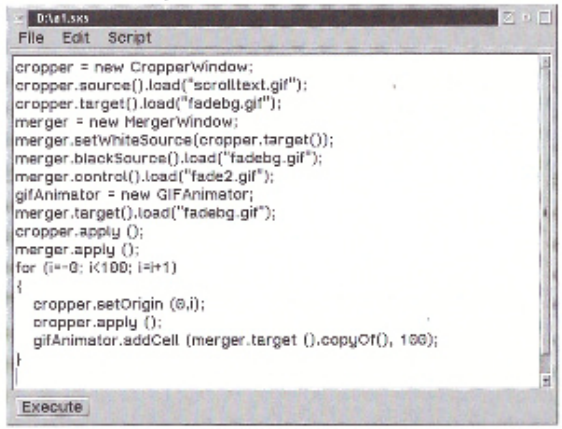
Chapter 11: WebAK Script

Scripts contain simple commands which instruct WebAK to perform tasks. By using scripts, you can create list of actions; A list which you can easily modify. Scripts are ideal for automating repetitive tasks.

WebAK Script has been designed to satisfy the needs of the casual user as well as computer scientists. In its basic form, you can use simple script commands to tell objects which you interact within WebAK's display to perform actions. For example, if you can use an Editor Window to load an image, you can use WebAK Script to tell the Editor Window to load the image. Generally, there is only one line per command. In its more advanced form, the WebAK Script evolves into a C++ style object oriented programming language. This section only covers the basic WebAK script; to do more complex tasks, please obtain the *WebAK Programmer's Guide* from Modular Dreams Incorporated homepage on the Internet at http://www.modulardreams.com.

The Script Window

The Script Window can be opened using the **Utilities>Script...** menu item. This basic window contains a text area where you can enter your script as well as some button and menu items to let you load and save scripts or control how their instructions are executed.



Script commands

All script commands take the same format:

Object . Method (Parameters) ;

Object - An object is the WebAK window which is to perform an action. For example: EditorWindow. You can derive the name of an object by using what is shown in its title bar, removing all spaces, attaching all the words and in some cases adding the word Window to the end. For example **MD+F WebAK Brightness** will become **BrightnessWindow** and **MD+F WebAK Histogram Equalization** becomes **HistogramEqualizationWindow**.

Method - A method is the action you would like the **Object** to perform. For example, the method "load" when used with an Editor Window tells it to load a file. The command to tell an Editor Window to load a file is *EditorWindow.load ("test.gif");*

Parameters - Parameters are the information the action needs to do its job. In the example *EditorWindow.load ("test.gif");* the parameter is the file name "test.gif". If there are no parameters, you should only use empty parenthesis and if there is more than one, they should be separated by commas.

Object identifiers

When you specify an object name such as EditorWindow or GIFAnimator, you are specifying them object by its class. It is perfectly fine to do this if there is only object of that class open but what if there are two, like two Editor Windows? If you use EditorWindow for the object name, your actions will be performed by the last EditorWindow you opened. If you wish to perform actions on both, you will need to give them names. An example of names for Editor Windows would be editorWindowOne and editorWindowTwo. The next section talks about how objects are created, and optionally, assigned names.

Creating objects

To create objects use the "new" command:

new Object;

An example to open an Editor Window and load an image into it is:

```
new EditorWindow;
EditorWindow.load ("stars.gif");
```


To perform the same action as above while giving the object its own unique name, you would need to do this:

```
starsEditorWindow = new EditorWindow;
starsEditorWindow.load ("stars.gif");
```

Destroying objects

To destroy objects use the following command:

delete Object;

An example to close open Editor Window is:

delete EditorWindow;

and if the Editor Window has a name, you can use it such as:

delete starsEditorWindow;

Finding out an object's methods and parameters

To find out the actions which an object can perform enter the following command:

Object.help ();

where Object is a class name. For example, if you want to know that actions you can request an Editor Window to perform, you would use:

EditorWindow.help ();

and the EditorWindow class would list you all the methods which it understands and is capable of performing.

Parameter types

As already mentioned, you can have methods which require no parameters, a single parameter, or a number of parameters separated by commas. When you request an object to list its methods, you will be presented with a list of the methods and the type of parameters they accept. There are many types of parameters (discussed in the *MD+F WebAK Programmer's Guide*) but the ones you will most frequently encounter are:

Strings - An string is a number of characters enclosed in double quotes, for

example "test.gif" is a string.
Integers - An integer is a whole number. For example 200 is an integer.
Floats - A float is any type of a number. Examples of floats are 200 and 200.5.
ImageKeepers - Imagekeepers hold the data which represents an image. (i.e. image bits, its dimensions, file name, format, etc.)

Script writing tips

The best way to learn WebAK Scripts is finding an already written script which performs an action similar to what you want to perform and studying its content. This way, you can not only see how the actions are performed, but you can also make small changes to them to learn even more. All scripts (.sxs SX script files) provided on Modular Dream Inc.'s homepage are commented to make it easier for them to be examined and understood.

Index

Installation Instruction

To install:
1. Insert the 1st diskette into the computer's floppy drive.
2. Open a Command session.
3. Run the INST script on the diskette.
4. Follow on screen instructions and switch diskettes when prompted to do so.

To execute:
Once installation is compete, drag the applications icon to your
desktop. Make sure you create shadow or a short cut to it, not move it. Double click to start the application.

Process Commander users: Do not use Process Commander to terminate WebAK. Doing so may freeze the computer in such a way that you will need to power it off and on. If this happens, CHKDSK will need to run on all your harddrives.

SX Paint users: A special version of MD+F SX Paint is included with WebAK 1.1 at no cost. This special version of SX Paint incorporates WebAK features into it but please note the following:
1 - WebAK does not support operation slicing. These options are overwritten in this version of SX Paint. In addition, the Airbrush and Erase features of SX Paint are currently disabled (accessible by using the right mouse button on the Freehand button on the Drawing Tools toolbar).
2 - WebAK does not require Fix Pack 1 for OS/2 Warp 4. SX Paint does.
3 - On-line help is only available in WebAK. Please visit Modular Dreams website for upcoming information pages on SX Paint.
4 - WebAK's collapse and resurrect feature as well as WebAK script automatically expand to support the Operation Windows (filters and effects) available in SX Paint, however, you should not depend on this feature since they are foreign to WebAK if someone uses your scripts or collapsed environment files with it.
5 - SX Paint puts OS/2 Warp's advanced multitasking to great use. If you are going to be creating a large number of tasks using it, please make sure the THREAD=Number and FILES=Number statements in your CONFIG.SYS are large enough to support SX.

SX Paint users: We would love to hear any suggestions you may have to improve this product. Please use the forms at **http://www.modulardreams.com/sx_development.html** which allows us to properly maintain the issues and suggestions you provide. A demo of SX Paint was released on September 18th 1996 and it has only been used for MD+F internal use until July 1st 1996. We expect to be making major upgrades to SX Paint on the MD+F Webpage in the near future.

Thank you for supporting OS/2 and OS/2 developers.